Date=27/07/2020
Lecture By=Shubham Joshi
Subject ⇒ Backtracking

IN PREVIOUS LECTURE **(QUICK RECAP)** **Date-24/07/2020**	**In Today's Lecture (Overview)**
Lower Case Upper Case	⇒ Backtracking In python ⇒ Question That Are Based on BackTracking ⇒ Mcq's ⇒ Questions For Self Practice / CC For the Day

⇒ Backtracking In python

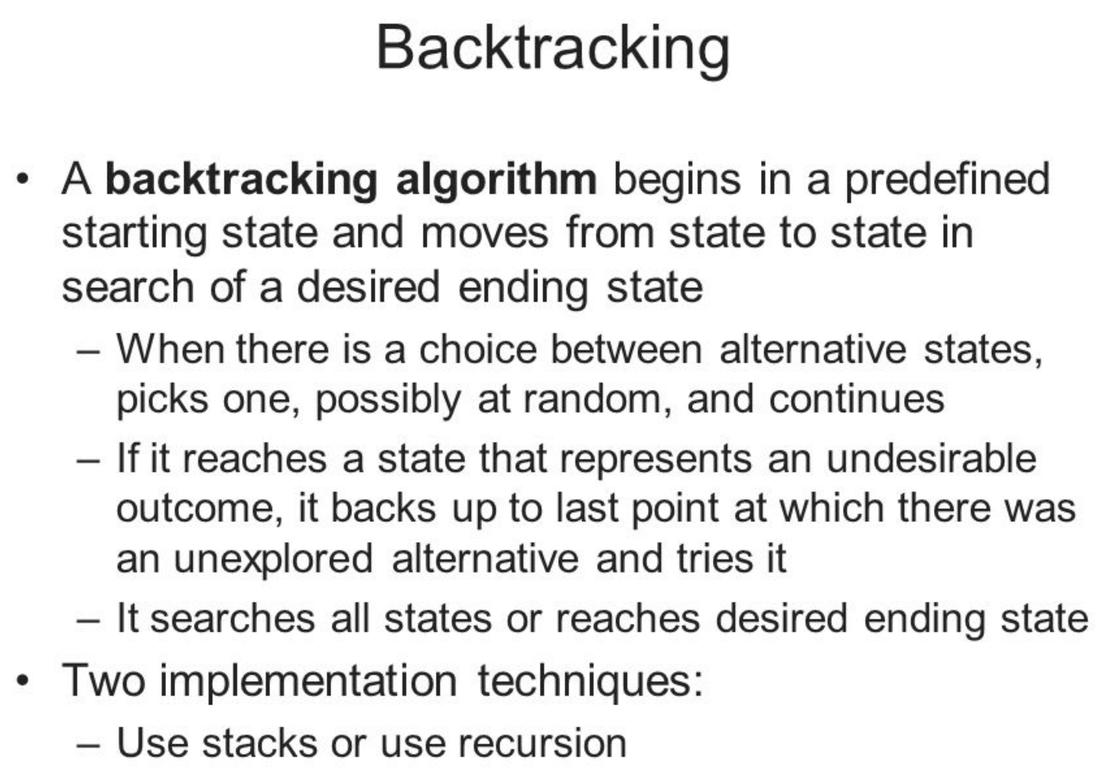
Backtracking is a form of **recursion**. But it involves choosing only option out of any possibilities.

We begin by choosing an option and **backtrack** from it

Backtracking is **an algorithmic-technique for solving problems recursively** by trying to build a solution **incrementally**, one piece at a time

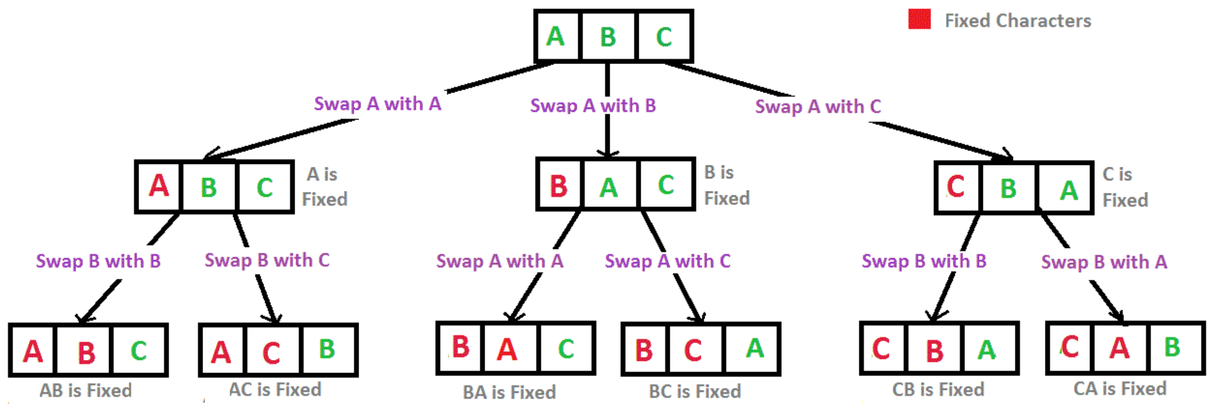
In Short
Backtracking builds a solution **incrementally**

"Click Here" To Know More About It



Backtracking

- A **backtracking algorithm** begins in a predefined starting state and moves from state to state in search of a desired ending state
  – When there is a choice between alternative states, picks one, possibly at random, and continues
  – If it reaches a state that represents an undesirable outcome, it backs up to last point at which there was an unexplored alternative and tries it
  – It searches all states or reaches desired ending state
- Two implementation techniques:
  – Use stacks or use recursion



Recursion Tree for Permutations of String "ABC"

*This Image is just for **Example Purpose**

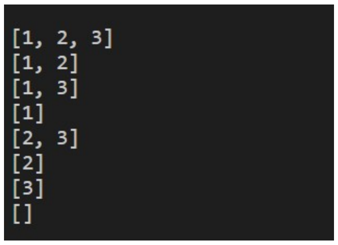
⇒ Question That Are Based on BackTracking

1.Print all Subsets Of 123 By Backtracking

Code

```python
def solve(a, idx, res):
        if idx == len(a):
                print(res)
                return
        res.append(a[idx])
        solve(a, idx + 1, res)

        res.pop()
        solve(a, idx + 1, res)

if __name__ == '__main__':
    solve([1,2,3], 0, [])
```

Output

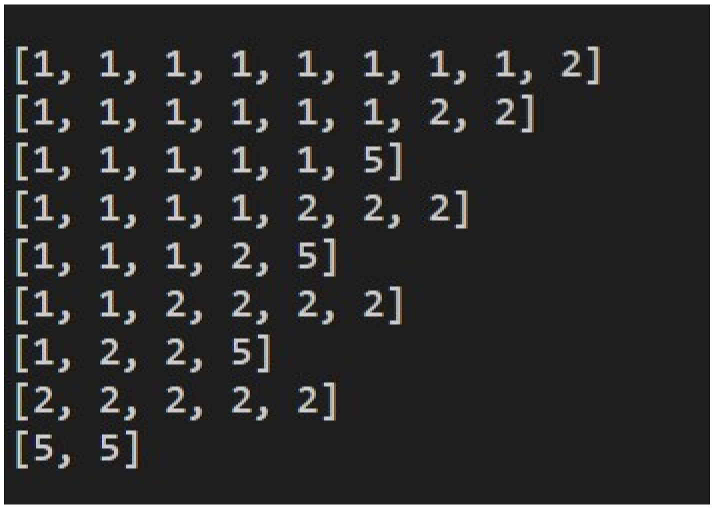


```
[1, 2, 3]
[1, 2]
[1, 3]
[1]
[2, 3]
[2]
[3]
[]
```

Question **2 = You are given a list [1,2,5] sum = 10**
You Have to print all Ways To Get Total Sum = 10

```python
def solve(a, sum, idx, res):
        if sum < 0:
                return
        if sum == 0:
                print(res)
                return

        if idx >= len(a):
                return

        res.append(a[idx])
        sum -= a[idx]
        solve(a, sum, idx, res)

        sum += a[idx]
        res.pop()
        solve(a, sum, idx + 1, res)

if __name__ == '__main__':
        solve([1,2,5], 10, 0, [])
```

Output



```
[1, 1, 1, 1, 1, 1, 1, 1, 2]
[1, 1, 1, 1, 1, 1, 2, 2]
[1, 1, 1, 1, 1, 5]
[1, 1, 1, 1, 2, 2, 2]
[1, 1, 1, 2, 5]
[1, 1, 2, 2, 2, 2]
[1, 2, 2, 5]
[2, 2, 2, 2, 2]
[5, 5]
```

⇒ Mcq’s

1.What is the complexity of coin change problem ?

(A)nlogn

(B)n

(C)1

(D)2^n

2.In backtracking which of the following is true ?

(A)you revisit the previous state

(B)you are ignoring previous state

3.What is the best time complexity for insertion sort ?

(A)n^2

(B)n

(C)1

(D)nlogn

4.out of bubble sort and selection sort Which has least no of swaps

(A)bubble

(B)Selection

(C)both have same

Answers
1.D
2.A
3.B
4.B

⇒ Questions For Self Practice / CC For the Day

https://practice.geeksforgeeks.org/problems/subsets/0
https://practice.geeksforgeeks.org/problems/combination-sum/0